CONTROL DATA CORPORATION

KRONOS® 2.1 APPLICATIONS PROGRAMMER'S INSTANT

CONTROL DATA®
CYBER 70 SERIES
MODELS 72, 73, 74
6000 SERIES
COMPUTER SYSTEMS

RECORD of REVISIONS

REVISION	NOTES
A	Manual released.
(12-73)	
B	Revised to reflect the KRONOS 2.1.2 operating system
(8-75)	at corrective code level 404. All systems-oriented in-
	formation has been removed and is now available in the
	KRONOS 2.1 Systems Programmer's Instant (publication
	no. 60449100). New features, as well as changes, de-
	letions, and additions to information in this manual, are
	indicated by bars in the margins or by a dot near the
	page number if the entire page is affected. This edition
	obsoletes all previous editions.

AA 3973

Publication No. 60407200

Printed in the United States of America

Address comments concerning this manual to:

Control Data Corporation
Publications and Graphics Division
4201 Lexington Avenue North
Arden Hills, Minnesota 55112

PREFACE

The KRONOS 2.1.2 Time-Sharing System provides network capabilities for time-sharing and transaction processing, in addition to local and remote batch processing on CONTROL DATA® CYBER 70 Series, Model 72, 73, and 74 Computer Systems, and CDC® 6000 Series Computer Systems.

This manual provides condensed descriptions of system control statements, control language formats, and loader, product set, and system utility control statement formats. Character set tables are also provided.

For descriptions of console commands, systems-oriented control statements, central memory tables, function requests, and external function codes, refer to the KRONOS 2.1 Systems Programmer's Instant.

The following manuals provide detailed descriptions of these subjects.

Control Data Publication	Publication No.
KRONOS 2.1 Systems Programmer's Instant	60449100
KRONOS 2.1 Reference Manual, Volume 1	60407000
KRONOS Terminal User's Instant	60407800
Loader Reference Manual	60344200
Loader Instant	60372200
Modify Reference Manual	60281700
Modify Instant	60283000
Update Reference Manual	60342500
Update Instant	60360200
ALGOL Reference Manual	60329000
ALGOL Instant	60192500
BASIC Reference Manual	19980300
COBOL Reference Manual	60384100
COBOL Instant	60328400
FORTRAN Extended Reference Manual	60305600
FORTRAN Extended Instant	60357900
SIMULA Reference Manual	60234800
Sort/Merge Reference Manual	60343900

CONTENTS

SYSTEM CONTROL STATEMENT FORMATS

PERMANENT FILE OPTIONS

The following control statement parameters are options on various permanent file commands.

Parameter	Description
UN=usernum	Specifies alternate user number for file residing in another user's catalog.
PW=passwrd	Specifies a 1- to 7-character password that must be specified whenever alternate users access the file.
CT=ct	Specifies category of permission for alternate users.

ct	Description
P or PRIVATE	Private files available for access only by originator or those with explicit permission
S or SPRIV	Semiprivate files available for access by users who know file name, user number, and password
PU or PUBLIC or LI	Public files available for access by all users who know file name, user number, and password

M=m — Specifies file or user permission mode.

m	Description
W or WRITE	Allows the user to write, read, append, execute, modify, or purge the file
M or MODIFY	Allows the user to modify, append, read, or execute a direct access file
A or APPEND	Allows the user to append information to the end of the file

Parameter	Description

m	Description
R or READ	Allows the user to read or execute the file
RM or READMD	Allows the user to read or execute a direct access file while another user is accessing the file in modify mode
RA or READAP	Allows the user to read or execute a direct access file while another user is accessing the file in append mode
E or EXECUTE	Allows the user to execute the file
N or NULL	Removes permission previously granted with the PERMIT control statement

R=r

Specifies the type of device on which the permanent file resides or is to reside.

r	Description
DA	6603 Disk System
DB	6638 Disk System
DC	863 Drum Storage
DDi	854 Disk Storage Drive ($1 \leq i \leq 8$)
DE	Extended core storage
DF	814 Disk File
DH	821 Data File
DIi	844 Disk Storage Subsystem ($1 \leq i \leq 8$)
DP	Distributive data path to ECS
MDi	841 Multiple Disk Drive ($1 \leq i \leq 8$)

S=space

Specifies the amount of space in PRUs desired for a direct access file.

PN=packname

A 1- to 7-character pack name used in conjunction with the R keyword to identify the device to be accessed in a permanent file request.

NA	Job does not abort if permanent file request fails.

TAPE MANAGEMENT OPTIONS

The following control statement parameters and keywords may appear on various tape management control statements.

Parameter	Description
D=den	Specifies tape density.

den	Description
LO or 200	200 bits per inch (bpi) (7-track)
HI or 556	556 bpi (7-track)
HY or 800	800 bpi (7-track)
HD or 800	800 characters per inch (cpi) (9-track)
PE or 1600	1600 cpi (9-track)

The keywords LO, HI, HY, HD, and PE may be specified instead of D=den.

FC=fcount	Specifies maximum block size in frames that may be read or written.
C=ccount	Specifies maximum size block in 6-bit characters that may be read or written.
CV=conv or N=conv	Specifies conversion mode for 9-track tapes.

conv	Description
AS	ASCII/display code conversion
US	Same as AS
EB	EBCDIC/display code conversion

MT	Specifies 7-track tape.
NT	Specifies 9-track tape.
PO=$p_1p_2,\ldots,p_n$	Specifies processing options.

Parameter	Description

P_i	Description
A	Abort job on irrecoverable read or write parity error
N	Do not abort job on irrecoverable read or write parity error
R	Enforce ring out
W	Enforce ring in
U	Inhibit unload
F	Force unload
E	Ignore all hardware read/write errors
B	Write system noise blocks when performing write error recovery
I	Ignore block being read when end of tape is encountered
P	Accept block being read when EOT is encountered
S	Specifies where system is to stop on an exit condition

F=format — Specifies data format.

format	Description
I	Internal
X	External
B	Blocked
E	Line image
S	Stranger tape
L	Long block stranger tape
SI	SCOPE internal
F	Foreign

NS=ns — Noise size.

LB=ℓ — Specifies whether a tape is to be treated as labeled or unlabeled.

ℓ	Description
KU	Unlabeled
KL	Labeled
NS	Nonstandard labels

VSN=vsn — A 1- to 6-character volume serial number that uniquely identifies a reel of tape.

Parameter	Description
CK	Specifies that lfn is to be used as a checkpoint file with information written at previous end-of-information (EOI).
CB	Specifies that lfn is to be used as a checkpoint file with information written at beginning-of-information (BOI).
F1=fileid or L=fileid	A 1- to 17-character file identifier.
FA=fa	File accessibility. If FA=A, only the owner of the tape can access the file. For other fa, all future accesses must specify the character as the fa parameter. FA omitted implies unlimited access.
OFA=fa	One character that indicates the current file accessibility of a labeled tape which is to be blank labeled (refer to FA description for explanation of fa).
S1=setid or M=setid	1- to 6-character set identifer for a multifile set.
SN=secno or V=secno	1- to 4-digit file section number.
QN=seqno or P=seqno	1- to 4-digit file sequence number.
G=genno	1- to 4-digit generation number.
E=gvn	1- to 2-digit generation version number.
CR=cdate or C=cdate	Creation date in form yyddd.
RT=rdate	Retention date in form yyddd.
OWNER=usernum/familyname	Identifies the owner of a labeled tape.
LSL=lsl	Label standard level. If LSL=1, the labels and data format are ANS1 standard. If omitted, indicates that format requires agreement of the interchange parties.

Parameter	Description
LO=ltype	Specifies the type of labels to list.

ltype	Description
A	Lists all required and optional ANSI labels
R	Lists all required labels
O	Lists all optional labels
V	Lists all VOL1-9 labels
H	Lists all HDR1 labels
F	Lists all EOF1-9 labels
E	Lists all EOV1-9 labels
U	Lists all UVL1-9 labels

L=out	Specifies the file on which the labels are to be listed.
U	Unload tape after blank labeling.
T=retcycle	1- to 3-digit retention cycle specifying number of days file is to be retained.
R	Directs the system to read an existing ANSI label.
W	Directs the system to write standard ANSI labels.
VA=va	Volume accessibility; one character specifying restrictions on who has access to information on the reel.

SYSTEM CONTROL STATEMENTS

APPEND (pfn, lfn_1, lfn_2, . . . , lfn_n/PW=passwrd, UN=usernum, PN=packname, R=r, NA)	Copies local files lfn_1 through lfn_n to end of indirect access permanent file pfn. †
ASCII.	Changes a time-sharing terminal to ASCII mode.

†Some parameters of this control statement are defined in Permanent File Options in this section.

ASSIGN(nn,lfn, D=den, {FC=fcount C=ccount}, CV=conv, {MT NT}, PO=$p_1p_2,\ldots,$ pn, F=format,NS=ns, LB=ℓ, VSN=vsn, {CK CB})	Assigns file lfn to the device or device type specified by nn. †
ATTACH(lfn$_1$=pfn$_1$,lfn$_2$=pfn$_2$, ...,lfn$_n$=pfn$_n$/ UN=usernum, PW=passwrd, M=m,PN=packname,R=r,NA)	Attaches permanent files pfn$_1$ through pfn$_n$ as local files lfn$_1$ through lfn$_n$ for direct access. †
BKSP(lfn,n,m)	Backspaces file lfn n logical records. m=C for coded mode, m=B for binary.
BLANK(D=den, {MT NT},VSN=vsn, FA=fa,VA=va, OWNER=usernum/ familyname, LSL=lsl,U)	Blank labels a magnetic tape. †
CATALOG(lfn, $p_1,p_2,\ldots,p_n$)	Catalogs file lfn.

p_i	Description
N=0	Catalog until an empty file is encountered.
N=x	Catalog x files; default is 1.
N	Catalog to end of information.
L=fname	Specifies output file.
U	Select user library list.
CS	Suppress character set list for OPL/OPLC type records.

†Some parameters of this control statement are defined in Permanent File Options or Tape Management Options in this section.

	p_i	Description
	D	Suppress comment field and page heading following first 1.
	R	Rewind lfn before and after cataloging.

CATLIST(LO=p, FN=pfn, UN=usernum, PN=packname, R=r, L=lfn, NA, DN=dn)

Lists information about user's permanent files and permanent files he can access in catalogs of alternate users. †

Options	Description
LO=F	Selects listing of pertinent information about each file in the user's catalog
LO=FP	Selects listing of permission information recorded for each alternate user of a specified file
LO=0	Selects a short list that includes only the names of the files in the user's catalog (this value assumed if LO omitted)
LO=P	Selects a short list that indicates the user numbers of alternate users who have accessed the specified file
FN=pfn	Permanent file name
L=lfn	Output file name (default is OUTPUT)
DN=dn	Device number

CHANGE(nfn=ofn/CT=ct, M=m, PW=passwrd, PN=packname, R=r, NA)

Allows originator of a permanent file to alter any of several parameters. If nfn=ofn is specified, file name ofn in the user's catalog is changed to nfn. †

CHARGE(chargenum, projectnum)

Specifies user's charge and project numbers for user profile control validation.

†Some parameters of this control statement are defined in Permanent File Options in this section.

CKP(lfn_1, lfn_2, ..., lfn_n)

Directs system to take a checkpoint dump; each lfn_i is included in the dump.

CLEAR.

Allows the user to return all files from his job.

COMMENT. comments
or
*comments

Enters comments in system and user's dayfile.

COMMON(lfn_1, lfn_2, ..., lfn_n)

Accesses a file that was already assigned common (library file type) status or assigns a local locked file to common status.

CONVERT(p_1, p_2, ..., p_n)

Converts text files to 64-character set.

p_i	Description
P=lfn_1	Input on file lfn_1 (default is OLD)
N=lfn_2	Output on file lfn_2 (default is NEW)
RS=n	Maximum record size in characters; $1 \leq n \leq 500$ (default is 300)
64	Convert from 63- to 64-character set; if omitted, no conversion takes place
TS=t_1	Convert from old to new time-sharing character set with terminal type t_i:

t_i	type
TTY	ASCII code with standard print
COR	Correspondence code with standard print
COR-APL	Correspondence code with APL print
MEM-APL	Memorex® 1240 with APL print
BLK-EDT	ASCII code with standard print block edit mode

R	Rewind input and output files prior to processing

	P_i	Description
	RC=m	Convert m decimal records (if omitted, m=1 assumed)
COPY(lfn_1,lfn_2,x,C)		Copies lfn_1 to lfn_2. If x is present, files are rewound before copy and rewound, verified, and rewound after copy. If C is present, copy an Sl, S, or L format coded tape to coded line format.
COPYBF(lfn_1,lfn_2,n,C)		Copies n binary files beginning at current position of lfn_1 to lfn_2. If C is present, copy an Sl, S, or L format coded tape to coded line format.
COPYBR(lfn_1,lfn_2,n,C)		Copies n binary records beginning at current position of lfn_1 to lfn_2. If C is present, copy an Sl, S, or L format coded tape to coded line format.
COPYCF(lfn_1,lfn_2,n,fchar,lchar)		Copies n coded files beginning at current position of lfn_1 to lfn_2. Portion of each line image to copy is specified by fchar (first character position) and lchar (last character position).
COPYCR(lfn_1,lfn_2,n,fchar,lchar)		Copies n coded records beginning at current position of lfn_1 to lfn_2. Portion of each line image to copy is specified by fchar and lchar.
COPYEI(lfn_1,lfn_2,x,C)		Copies lfn_1 (current position to EOI) to lfn_2. If x is present, files are rewound before copy and rewound, verified, and rewound after copy. If C is present, copy an Sl, S, or L format coded tape to coded line format.
COPYSBF(lfn_1,lfn_2,n)		Copies n coded files beginning at current position of lfn_1 to lfn_2, shifting each line image one character to the right and adding a leading space.
COPYX(lfn_1,lfn_2,x,b,C) or COPYX(lfn_1,lfn_2,type/name,b,C)		Copies logical records from lfn_1 to lfn_2 beginning at current position of lfn_1 and continuing until terminator specified by x or type/name is encountered. Files are then backspaced according to b parameter. If C is present, copy an Sl, S, or L format coded tape to coded line format.

x — Terminator type:

- 00 — Zero record
- n — n records (default is 1)

type/name — Record name
name is first 7 characters of record; type is:

- ABS — Multiple entry point overlay
- COS — Chippewa format CP program
- OPL — Modify OPL deck
- OPLC — Modify OPL common deck
- OPLD — Modify OPL directory
- OVL — CP overlay
- PP — 6000 series PP program
- PPU — 7600 PP program
- REL — Relocatable CP program
- TEXT — Unrecognizable as a program
- ULIB — User library program

b — Backspace control:

- 0 — No backspace (default)
- 1 — Backspace lfn_1
- 2 — Backspace lfn_2
- 3 — Backspace lfn_1 and lfn_2

CSET,m. — Changes a time-sharing terminal's character set to m (ASCII or NORMAL).

CTIME. — Enters accumulated CPU time in system and user's dayfile.

DAYFILE(lfn) — Write user's dayfile on lfn; default is OUTPUT.

DEFINE(lfn$_1$=pfn$_1$,lfn$_2$=pfn$_2$,...,lfn$_n$=pfn$_n$/PW=passwrd,CT=ct,M=m,R=r,S=space,PN=packname,NA) — Creates an empty direct access permanent file or defines an existing local file as a direct access file. †

†Some parameters of this control statement are defined in Permanent File Options in this section.

DISPOSE(lfn_1=q_1, lfn_2=q_2,...,lfn_n=q_n/ot=usernum)

Releases files to specified output queues.

q_i	Queue type
PR	Print
PH	Punch coded 026
P9	Punch coded 029
PB	Punch binary
P8	Punch 80-column format

The origin types are specified with the ot parameter where BC is local batch origin and EI is remote batch origin. The number of the remote batch (EI) user is specified with usernum.

DMD(fwa,lwa)
or
DMD(lwa)
or
DMD.

Dumps central memory from first word address to last word address minus 1; output contains display code equivalences. If lwa alone is present, fwa=0 is assumed. If neither fwa nor lwa is present, DMD dumps exchange package and 40_8 locations before and after program address in exchange package.

DMP(fwa,lwa)
or
DMP(lwa)
or
DMP.

Dumps central memory from first word address to last word address minus 1. If lwa alone is present, fwa=0 is assumed. If neither fwa nor lwa is present, DMP dumps exchange package and 40_8 locations before and after program address in exchange package.

DOCMENT(p_1,p_2,...,p_n)

Enables the user to extract the external or internal documentation from a file containing COMPASS source code.

p_i	Description
I=lfn_1	Name of file that contains page footing information in following format:

Column	Contents
1	Blank
2-45	Document title
46-55	Publication number
56-60	Revision level
61-70	Revision date

S=lfn$_2$	Name of file containing source statement images
L=lfn$_3$	Name of file to receive output
N=nn	Number of copies
T=type	Documentation type (INT for internal or EXT for external)
C=cc	Key character for documentation
P=pp	Number of print lines per page
NT	Negate table generator
TC	List table of contents

ENQUIRE($p_1 p_2$,...,p_n) Lists information about a user's job specified by the options.

p_i	Description
OP=A or A	Causes all OP= options to be processed
OP=B or B	Returns information concerning user identification and priorities
OP=F or F	Status of files at the user's control point
OP=J or J	Returns contents of control registers and error flag field
OP=L or L	Returns user's loader information
OP=R or R	Returns system resources used
OP=S or S	Returns SRUs
OP=T or T	Returns accumulated CPU time
OP=U or U	Returns amount of resources available to the user
JN=jnm	Returns status of remote batch job jnm (last three characters of name assigned by system) initiated with SUBMIT command

	P_i	Description
	FN=lfn_1	Returns status of file lfn_1
	O=lfn_2	Specifies file to receive output (default is OUTPUT)
	If no parameters are specified, default is OP=A.	
EVICT(lfn_1, lfn_2,...,lfn_n)	Releases file space for lfn_i but does not release the file attachment to the job.	
EXIT.	Indicates where in control statement record to resume control statement processing if an error is encountered or where to terminate normal control statement processing.	
FAMILY (familyname)	Allows user to change the family name associated with his job.	
GET(lfn_1=pfn_1, lfn_2=pfn_2,..., lfn_n=pfn_n/UN= usernum, PW= passwrd, PN= packname, R=r, NA)	Retrieves a copy of indirect access permanent file pfn_i for use as a local file lfn_i. †	
GTR(lfn_1, lfn_2, D, NR, S) selection directives	Copies records specified by selection directives from lfn_1 to lfn_2, starting at current EOI of lfn_2.	
	D	Causes a directory record to be written at the end of lfn_2
	NR	Specifies that files lfn_1 and lfn_2 are not rewound before or after the operation
	S	Processes lfn_1 as a sequential file
	selection directives	Description
	type/name	Retrieves record of specified type and name
	name	Retrieves record specified

†Some parameters of this control statement are defined in Permanent File Options in this section.

	selection directives	Description
	0	Inserts a zero-length record on file lfn_2
	type/$name_1$ -$name_2$	Retrieves records $name_1$ through $name_2$ of type specified

jobname(Tt, CMfl, Pp)
or
jobname(t, fl, p)

Specifies name, time limit, field length, and priority of job.

KRONREF(P=lfn_1, L=lfn_2, S=lfn_3, G=lfn_4)

Generates a cross-reference listing of system symbols used by decks on a MODIFY OPL.

P=lfn_1	OPL input on file lfn_1 (default is OPL)
L=lfn_2	List output on file lfn_2 (default is OUTPUT)
S=lfn_3	System text from overlay lfn_3 (default is SYSTEXT)
G=lfn_4	System text from local file lfn_4 (default is TEXT)

LABEL(lfn, D=den, FC=fcount, CV=conv, {MT / NT}, PO=p_1p_2, ..., p_n, F=format, NS=ns, LB=ℓ VSN=vsn, {CK / CB}, {FI=fileid / L=fileid}, FA=fa, {SI=setid / M=setid}, {SN=secno / V=secno}, {QN=seqno / P=seqno}, G=genno, E=gvn, {CR=cdate / C=cdate}, {RT=rdate / T=retcycle}, {W / R})

Assigns lfn to a tape unit and creates a new or accesses an existing tape. †

†Some parameters of this control statement are defined in Tape Management Options in this section.

LBC(addr)

Loads binary corrections, beginning at addr, into central memory.

LDI(lfn, id)

Copies batch job image on lfn to mass storage and submits it to the input queue with identifier id.

LENGTH(lfn)

Returns status of file lfn.

LIBGEN($p_1, p_2, \ldots, p_n$)

Generates a user library file.

p_i	Description
$F=lfn_1$	Name of source file containing records to be placed on user library file lfn_2 (default is LGO)
$P=lfn_2$	Name of file on which the library is to be written (default is ULIB)
$N=lfn_3$	Name of user library being generated (default is lfn_2)
NX=n	If n is nonzero, no cross-references are given (default is n=0)

LIMITS

Lists validation information for user named on current USER statement.

LINK($p_1, p_2, \ldots, p_n$)

Specifies directives for the LINK loader.

p_i	Description
$F=lfn_1$	Loads from file lfn_1 (default is LGO)
$P=lfn_2$	External references on program library lfn_2 (default is SYSLIB)
$B=lfn_3$	Write loaded program on file lfn_3
$L=lfn_4$	Write load map on file lfn_4 (default is OUTPUT)
E=name	Load program with specified entry point name from file lfn_1

LO=chars	Set map option S for statistics, errors, and any of the following: B Block assignments E Entry points X External references and entry points
X	Execute loaded program

LISTLB(lfn, {SI=setid / M=setid}, {QN=seqno / P=seqno}, LO=ltype, L=out)	Reads ANSI labels on file lfn and writes them on file specified by out. †
LIST80(lfn_1, lfn_2, NR)	Reads file lfn_1 containing COMPASS source code and writes it, compressed to 80 columns, on lfn_2. NR specifies that lfn_1 is not rewound.
LOC(fwa,lwa) or LOC(lwa) or LOC.	Enters octal correction statement images from INPUT into central memory in specified area.

†Some parameters of this control statement are defined in Tape Management Options in this section.

LOCK(lfn_1, lfn_2, ..., lfn_n)

Sets write interlock bit in FNT/FST entry for local file lfn_i.

LO72(p_1, p_2, ..., p_n)

Reformats files to 72 columns.

P_i	Description
I=lfn_1	Reformat parameters are on file lfn_1 (default is INPUT)
S=lfn_2	Data to be reformatted is on file lfn_2 (default is SCR)
L=lfn_3	Reformatted data is listed on file lfn_3 (default is OUTPUT)
H=xxx	Number of characters per output line up to 150 (default is 72)
LP	Output is formatted for line printer
NR	Output file is not rewound
Nx=y	Specifies number of characters to be moved (up to 6 fields): x(1 to 6) Number of field being moved y Number of characters being moved
Ix=y	Specifies the field the data originates from where x is as in Nx and y is starting column of originating field
Ox=y	Specifies the destination the data is going to where y is the starting column of destination field

MODE(m)	Sets CPU program exit mode to m ($0 \leq m \leq 7$).
NEW,lfn/ND.	Allows the user to create a new primary file. The old primary file and all local files are returned unless the ND keyword is specified.
NOEXIT.	Suppresses transfer to card following next EXIT statement if an error occurs.
NORERUN.	Clears rerun status of job.
OFFSW(s_1,s_2, ...,s_n)	Clears pseudo-sense switches for reference by user's program.
OLD,lfn/ND.	Allows the user to get the indirect access permanent file specified by lfn and make it the primary file. Any previous primary file is returned and all local files are returned unless the ND keyword is specified.
ONEXIT.	Reverses effect of NOEXIT statement.
ONSW(s_1,s_2, ...,s_n)	Sets pseudo-sense switches for reference by user's program.
OUT.	Releases output files from control point to the output queue.
PACK(lfn_1, lfn_2,x)	Packs lfn_1 into one record on lfn_2. If x is specified, lfn_1 is not rewound prior to pack.
PACKNAM (PN=packname) or PACKNAM (packname)	Directs subsequent permanent file requests to the specified auxiliary device.
PARITY,p.	Changes a time-sharing terminal's parity to p (ODD or EVEN).
PASSWOR(oldpswd,newpswd)	Changes user's password from oldpswd to newpswd.
PBC(fwa,lwa)	Writes one record from specified area in central memory on PUNCHB.

PERMIT(pfn, $username_1$=m_1, $username_2$=m_2,..., $username_n$=m_n/ PN=packname, R=r,NA)	Allows user to explicitly permit another user to access a private file in his permanent file catalog with permission m_i. †
PRIMARY,lfn.	Allows the user to return the current primary file and make lfn the primary file.
PURGALL(CT=ct, AD=ad,MD=md, CD=cd,DN=dn, TY=ty,TM=tm, PN=packname, R=r,NA)	Purges all permanent files in the user's catalog as specified by parameters. †

Parameter	Description
ct	File category
ad	Last access date
md	Last modification date
cd	Creation date
dn	Device number
ty	File type(INDIR, DIRECT, or ALL)
tm	Time of day on the date specified by ad, md, or cd

PURGE(pfn_1, pfn_2,...,pfn_n/ UN=usernum, PW=passwrd, PN=packname, R=r,NA)	Allows user to remove a file from the permanent file device. †
RBR(n,name)	Loads one binary record from a specified file. If n is less than four characters and is numeric, TAPEn is the file name. If n contains a nonnumeric character or is four or more characters long, n itself is the file name. If n is omitted, TAPE is the file name. name is a 1- to 7-character name used in a record prefix.
RENAME($nlfn_1$=$olfn_1$, $nlfn_2$=$olfn_2$,..., $nlfn_n$=$olfn_n$)	Changes name of file $olfn_i$ to $nlfn_i$ in FNT/FST.

†Some parameters of this control statement are defined in Permanent File Options in this section.

REPLACE(lfn$_1$=pfn$_1$,lfn$_2$=pfn$_2$,...,lfn$_n$=pfn$_n$/
UN=usernum,
PW=passwrd,
PN=packname,
R=r,NA)

Substitutes new file lfn$_i$ for old file pfn$_i$. †

REQUEST(lfn,
D=den,
{FC=fcount / C=ccount},
CV=conv, {MT / NT},
PO=p$_1$p$_2$,...,p$_n$,
F=format,NS=ns,
LB=ℓ
VSN=vsn, {CK / CB})

Requests operator to assign a device to lfn. †

RERUN.

Sets rerun status for job.

RESEQ(lfn,t,
xxx,yy)

Resequences source files that have leading sequence numbers.

lfn — Name of file to be sequenced

t — Type of file:

t	Description
B	BASIC source
T	Text source
other or omitted	Any number at beginning of line is considered sequence number

xxx — New line number of first statement

yy — Line number increment

†Some of the parameters of this control statement are defined in Permanent File Options in this section.

RESOURC(rt_1=u_1, rt_2=u_2,..., rt_n=u_n)

Specifies maximum number of tape units or disk packs.

rt_i	Description
MT	Magnetic tape (7-track)
NT	Magnetic tape (9-track)
DDi	854 Disk Storage Drive ($1 \leq i \leq 8$)
DIi	844 Disk Storage Subsystem ($1 \leq i \leq 8$)
MDi	841 Multiple Disk Drive ($1 \leq i \leq 8$)

The maximum number of units of resource type rt_i the job will use concurrently is specified with u_i.

RESTART(lfn, nnnn, x_i)

Restarts a previously terminated job from a specified checkpoint.

lfn	Checkpoint file
nnnn	Number of checkpoint from which to restart

x_i	Description
RI	Control statement file on lfn is not restored
NA	RESTART does not abort if a required file is not available
FC	If a file is local to restart job, RESTART does not replace it with the file on the checkpoint dump

RETURN(lfn_1, lfn_2,..., lfn_n)

Releases job attachment and/or file space of lfn_i.

REWIND(lfn_1, lfn_2,..., lfn_n)

Rewinds the files and positions them to BOI.

RFL(nnnnnn)

Changes job field length from that specified on the job card to nnnnnn.

ROLLOUT.

Rolls out user's job and releases all memory assigned to the job.

RTIME.

Issues current time in milliseconds to dayfile.

SAVE(lfn$_1$=pfn$_1$, lfn$_2$=pfn$_2$, ..., lfn$_n$=pfn$_n$/PW=password, CT=ct, M=m, PN=packname, R=r, NA)	Retains copy of local file lfn$_i$ as an indirect access file pfn$_i$. †
SETCORE(p) or SETCORE(-p)	Sets each word within the field length to the fill character specified by p. If -p, complement of p is set.
SETID(lfn$_1$=x$_1$, lfn$_2$=x$_2$, ..., lfn$_n$=x$_n$)	Assigns a new identification code x$_i$ for lfn$_i$.
SETPR(p)	Specifies a new CPU priority for user's job (may be increased only if job is system origin or contains SSJ= entry point).
SETTL(t)	Specifies a new time limit for user's job.
SKIPEI(lfn)	Positions lfn at EOI.
SKIPF(lfn, n, m)	Bypasses n files, in the forward direction, from current position on lfn. m=C for coded mode, and m=B for binary.
SKIPFB(lfn, n, m)	Bypasses n files, in the reverse direction, from current position on lfn. m=C for coded mode, and m=B for binary.
SKIPR(lfn, n, ℓ, m)	Bypasses n records, in the forward direction, from current position on lfn. ℓ specifies EOR level.
SORT(lfn, NC=n)	Sorts a file, lfn, of line or statement images in numerical order based on leading line numbers consisting of n digits.
STAGE(lfn, p$_1$, p$_2$, ..., p$_n$)	Causes files to be copied from specified device to mass storage file lfn.

p_i	Description
NR	Do not rewind lfn before operation

†Some of the parameters for this control statement are defined in Permanent File Options in this section.

	P_i	Description
	NU	Do not unload lfn after staging operation
	DR	Drop job after staging operation
	N=n	Copy n files to lfn
	T=xx	Stage lfn from device with EST ordinal xx
	VSN=vsn	1- to 6-character volume serial number of tape associated with lfn
	D=den	Tape density
	F=format	Data format (I, X, or SI)
	MT	7-track tape
	NT	9-track tape

STIME. Issues the current value of the SRU accumulator to the user's dayfile.

SUBMIT(lfn, q, NR)c Submits a batch job on lfn to the input queue for processing.

q		Specifies disposition of job output:
	B	Disposed to local batch queue and printed/punched at central site
	N	Disposed to local batch queue, dropped at job termination
	E	Disposed to remote batch queue, printed at remote batch terminal
NR		Inhibits rewind of file specified by cREAD
c		Escape character used to identify reformatting directives (if omitted, / is assumed)

Reformatting directives:

cJOB	Reformats submit file (selects cNOTRANS, cSEQ, and cPACK)
cEOR	Writes end-of-record
cEOF	Writes end-of-file
cSEQ	Removes subsequent line numbers
cNOSEQ	Reverses effect of cSEQ
cPACK	Removes subsequent EOR and EOF marks
cNO-PACK	Reverses effect of cPACK directive
cTRANS	Indicates transmission mode
cNO-TRANS	Reverses effect of cTRANS directive
cREAD, lfn	Inserts file lfn in place of cREAD directive in submit file
cRE-WIND, lfn	Rewinds file lfn to BOI
c_1EC=c_2	Changes escape code character from c_1 to c_2

SUI(n) Allows user to access a permanent file catalog without using a USER statement. n specifies a user index number (SYOT only).

SUMMARY. Lists the current resource usage for a job.

SWITCH(s_1, s_2,...,s_n) Sets the pseudo-sense switches for reference by the user's program.

TDUMP(p_1, p_2, ..., p_n) Lists a file in octal or alphanumeric form

p_i-	Description
I=lfn$_1$	Input file name (default is TAPE1)
L=lfn$_2$	Output file name (default is OUTPUT)

p_i	Description
O	Octal dump only (default is O and A)
A	Alphanumeric dump only (default is O and A)
R=rcount	Number of records to dump
F=fcount	Number of files to dump
N=lines	Maximum lines that can be dumped
NR	Do not rewind lfn_1 before dump

UNLOAD(lfn_1, lfn_2,...,lfn_n) Performs the same function as RETURN.

UNLOCK(lfn_1, lfn_2,...,lfn_n) Clears the write interlock bit for local file lfn_i.

UPMOD(p_1, p_2,...,p_n) Converts Update-formatted program library to a Modify-formatted program library file.

p_i	Description
P=lfn_1	Update program library from file lfn_1 (default is OLDPL)
N=lfn_2	Modify program library on file lfn_2 (default is OPL)
M=lfn_3	Modify program library name is lfn_3 (default is OPL)
F	Convert to file mark
NR	Do not rewind lfn_1

USECPU(n) Specifies which CPU is to be used for processing: CPU0 for n=1 and CPU1 for n=2.

USER(usernum passwrd, familyname) Sets validation and permanent file base for a user number.

usernum	User number
passwrd	User's password
familyname	Identifies family of permanent devices

VERIFY(lfn_1, lfn_2, p_1, p_2, ..., p_n)

Performs a binary comparison of all data from the current position of the files specified.

lfn_1	Name of first file (if omitted, TAPE1 assumed)
lfn_2	Name of second file (if omitted, TAPE2 assumed)

p_i	Description
N=0	Verify terminates on first empty file encountered on either file
N=x	Verify x files (default is 1)
N	Verify terminates when EOI is encountered on either file
E=y	List first y errors (if omitted, 100 assumed)
L=lfn_3	List errors on lfn_3 (default is OUTPUT)
A	Abort if errors occur
R	Rewind both files before and after

VFYLIB(lfn_1, lfn_2, lfn_3, NR)

Performs a comparison of binary records on files lfn_1 and lfn_2 and lists replacements, deletions, and insertions on lfn_3. If NR is specified, lfn_1 and lfn_2 are not rewound.

VSN(lfn_1=vsn_1, lfn_2=vsn_2, ..., lfn_n=vsn_n)

Associates volume serial number vsn_i with file lfn_i.

WBR(n, rl)

Writes a binary record of length rl from central memory on the specified file, beginning at its current position. Refer to RBR for description of n.

WRITE(lfn, x)

Writes x file marks on lfn.

WRITER(lfn, x)

Writes x empty records on lfn.

CONTROL LANGUAGE FORMATS

CALL(lfn, { C / S=ccc }, RENAME (oldnam$_1$=newnam$_1$, oldnam$_2$=newnam$_2$, ..., oldnam$_n$=newnam$_n$)
or
CALL(lfn, { C / S=ccc }, (oldnam$_1$=newnam$_1$, oldnam$_2$=newnam$_2$, ..., oldnam$_n$=newnam$_n$)

Inserts procedure file (lfn) at specified position in the control statement stream.

DISPLAY (expression)

Evaluates expression and displays result in the dayfile. Expression can be any legal control language expression.

FILE(lfn, expression)

Determines status of file lfn. expression is any legal expression. FILE expressions, however, use symbolic names.

Symbolic Names

Names with values:

EQ	Equipment status table (EST) ordinal (0 through 77_8)
ID	File ID (0 through 67_8)

Names with true/false values:

MS	File is on mass storage
LK	File is locked
OP	File is opened
EX	Execute-only file
AS	File is assigned to user's control point

File types:

LO	Local
CM	Common
PR	Print
IN	Input
PH	Punch
LI	Library
PM	Direct access permanent file
PT	Primary

Device types:

CP	415 Card Punch
CR	405 Card Reader
DA	6603 Disk System
DB	6638 Disk System
DC	863 Drum Storage
DD	853/854 Disk Storage Drive
DE	Extended core storage
DF	814 Disk File
DH	821 Data File
DI	844 Disk Storage
DP	Distributive data path
DS	Console display
LP	501, 505, 512, or 580 Line Printer
LQ	512 Line Printer
LR	580 Line Printer
MD	841 Multiple Disk Drive
MS	Mass storage
MT	Magnetic tape drive (7 track)
NE	Null equipment
NT	Magnetic tape drive (9 track)
ST	6671 Multiplexer
TT	Time-sharing multiplexer (6671 or 6676)

GOTO(stmt)

Transfers control to another location within the control statement file. stmt is name of any control statement or a digit (0 through 9) followed by up to six alphanumeric characters.

IF(expression)stmt.
or
IF(SS op ssname) stmt.
or
IF(SS op ssname expression)stmt.

If the conditions given in expression are true, stmt is processed. The expression is considered true if it is evaluated to a nonzero value.

stmt	Any legal control language statement
expression	Any legal expression

	op	One of the operators: = .EQ. ≠ .NE.
	ssname	Any legal subsystem name
NUM(name)	Determines if name has a numeric value.	
SET(Ri=expression) or SET(EF=expression) or SET(SS=ssname)	Allows user to specify a subsystem or set software registers to control flow of a job. Ri indicates software-defined register 1, 2, or 3 (18 bits). EF is error flag register (6 bits). The parameter ssname is any legal subsystem name.	

Symbolic Names Used in Expressions

Names with values:

R1	Contents of control register 1
R2	Contents of control register 2
R3	Contents of control register 3
FL	Job field length
EM	Current exit mode
EF	Previous error flag
TLE	Time limit error
ARE	Arithmetic error
PPE	PPU abort
CPE	CPU abort
MNE	Monitor call error
ODE	Operator drop
PSE	Program stop error
TKE	Track limit error
FLE	File limit error
OT	Job origin type
SYO	System origin
BCO	Local batch origin
EIO	Export/Import origin
TXO	Time-sharing origin

SS	Job subsystem: NULL BASIC FTNTS EXECUTE BATCH ACCESS TRANACT

Names with Boolean value:

SWn	Setting (1=on, 0=off) of sense switch n ($1 \leq n \leq 6$)
TRUE	True value
T	True value
FALSE	False value
F	False value

CYBER LOADER CONTROL STATEMENT FORMATS

EXECUTE (eptname, p_1, p_2, ..., p_n)	Causes completion of a load and execution of the loaded program.
eptname	Name of entry point in one of the loaded modules at which execution is to begin.
p_i	Execution-time parameters to be passed to the loaded program.
LDSET(option$_1$, option$_2$, ..., option$_n$)	Provides user with control of load operations.

option$_i$†	Description
LIB=libname$_i$	Specifies one or more libraries composing the local library set.
MAP=p_1/lfn$_1$ or MAP=/lfn$_1$ or MAP=p_1	Controls the generation of the load map. The MAP is written to file lfn$_1$. The map contents is specified by p. N No map S Statistics B Block map E Entry point map X Entry point cross-references
PRESET=p_2	Specifies the values to which unused core in central memory field length is set prior to execution of the loaded program.

p	Octal Preset Value
NONE	No presetting
ZERO	00...0
ONES	77...7
INDEF	177700...0
INF	377700...0
NGINDEF	600...0
NGINF	400...00addr
ALTZERO	2525...2525
ALTONES	5252...5252

†Multiple parameters for LDSET options are separated by slashes. For example, LIB=LIB1/LIB2/LIB3.

$option_i$	Description
ERR=p_3	Selects one of three methods of handling loader errors.
	p — **Significance** ALL — Program aborted for fatal, nonfatal, and terminal errors FATAL — Program aborted for fatal and terminal errors NONE — Terminal errors cause job abortion
REWIND and NOREWIN	Alters the default option for rewinding of files prior to loading
USEP=$pname_i$	Causes the indicated object modules to be loaded regardless of whether or not they are needed to satisfy external references.
USE=$eptname_i$	Forces the loading of object modules to ensure that specified entry points are included in the load.
SUBST=$pair_i$†	Changes external references to entry point names to other entry point names. $pair_i$ is a pair of entry point names in the form: $eptname_1$-$eptname_2$. As a result of SUBST, a reference to $eptname_1$ becomes a reference to $eptname_2$.
OMIT=$eptname_i$†	Directs that the specified entry point names are to remain unsatisfied, regardless of whether the module containing these entry point names is loaded.
FILES=lfn_i	Permits record manager users to ensure that library programs are loaded for the processing of specified files.

†Not available for programs loaded from a library generated with a cross-reference ULIB directory.

LIBLOAD(libname, eptname$_1$, eptname$_2$, ..., eptname$_n$	Performs load of modules from a library.
libname	Name of library containing object modules having the specified entry point names (eptname$_i$).
LOAD(lfn$_1$, lfn$_2$, ..., lfn$_n$)	Loads object modules.
lfn$_i$	Name of file to load.
NOGO(lfn, eptname$_1$, eptname$_2$, ..., eptname$_n$)	Causes completion of a load.
lfn	Name of logical file on which core image module is to be written.
eptname$_i$	Names of entry points to be included in header.
SATISFY(libname$_1$, libname$_2$, ..., libname$_n$)	Satisfies external references.
libname$_i$	Name of system or user library.
SLOAD(lfn, name$_1$, name$_2$, ..., name$_n$)	Requests loader to load modules from a local file.
lfn	Local file name.
name$_i$	Names of modules to be loaded in the order encountered on lfn.

SYSTEM UTILITY CONTROL STATEMENT FORMATS

LIBEDIT($p_1, p_2, \ldots, p_n$) Edits and replaces uniquely identifiable records on a file with records from one or more correction files.

p_i	Description
$I=lfn_1$	Directives comprise the next record on file lfn_1 (if omitted, INPUT assumed).
$P=lfn_2$	File lfn_2 contains the old program library (if omitted, OLD assumed).
$N=lfn_3$	New program library is written on file lfn_3 (if omitted, NEW assumed).
L=1	Short correction listing on file specified by LO parameter (if omitted, full correction listing).
$LO=lfn_4$	List output on file lfn_4 (if omitted, OUTPUT assumed).
$B=lfn_5$	Use file lfn_5 for the replacement file (if omitted, LGO assumed).
C	Copy the new library file over the old library file after processing.
R	Do not rewind library files after processing.
V	Call VFYLIB after LIBEDIT processing.
D	Ignore errors and continue.

The I, P, N, L, and B parameters are turned off by specifying p_i=0. If the C, R, V, or D parameters are omitted, the indicated action does not occur.

The following parameters are common to several LIBEDIT directives.

rid Specifies a reference point for a correction.

type/rname	Reference record is of specified type
rname	Reference record is the implied type
*	Reference point is an EOF (*BEFORE only)

gid

Indicates records or groups of records to be inserted, deleted, or replaced.

type/rname	Single record of the specified type
$type_1$/ $rname_1$- $type_2$/ $rname_2$	Group of records beginning with $rname_1$ of $type_1$ and ending with $rname_2$ of $type_2$ where $rname_i$ is a record identifier and $type_i$ is the type of the named record

Directive	Description
*ADD lib, gid_1, gid_2, . . . ,gid_n	Appends records to the specified library lib for transcription to the new library.
*BEFORE rid, gid_1, gid_2, . . . , gid_n	Inserts records from the current replacement file before the specified old library record for transcription to the new library file (*B also legal).
*BUILD dname	Constructs and appends a directory record in modify format to the new library file. dname specifies the name of the directory record.
*COMMENT rid comment	Adds a comment to the prefix table for a program on a replacement file or the old library file.
*COPY	Copies the new library file to the old library file after processing corrections.
*DATE rid comment	Adds the current date and specified comment (up to 40 characters) to the prefix table.
*DELETE gid_1, gid_2, . . . , gid_n	Supresses copying of specified records from the old library file to the new library file (*D also legal).
*FILE lfn	Declares a secondary file lfn that contains replacement records.

*IGNORE gid_1, gid_2, . . . , gid_n	Ignores records on the current replacement file during record processing.
*INSERT rid, gid_1, gid_2, . . . , gid_n	Inserts records from the current replacement file after the specified old library record for transcription to the new library file (*I, *A, and *AFTER also legal).
*NOREP lfn_1, lfn_2, . . . , lfn_n	Declares the specified replacement files lfn_i to be no-replace files.
*RENAME rid, name	Assigns a new name to a record on the old library or the current replacement file for transcription to the new library file.
*REPLACE gid_1, gid_2, . . . , gid_{nn}	Replaces records on the old library file with records of the same name from a current replacement file that has been declared a no-replace file.
*REWIND lfn	Rewinds file lfn before and after editing.
*TYPE type or *NAME type	Specifies default type of internal record format.

type	Description
REL	Relocatable CPU program
OVL	CPU overlay program
ABS	Multiple entry point overlay
PP	PPU program
PPU	7600 PPU program
OPL	Modify OPL deck
OPLC	Modify OPL common deck
OPLD	Modify OPL directories
ULIB	User library
COS	Chippewa format CPU program
TEXT	Unrecognizable as a program

MODIFY(p_1, p_2, ..., p_n) Calls the MODIFY program.

p_i	Description
I=lfn_1	Directive input on file lfn_1.
P=lfn_2	Old program library on file lfn_2.
C=lfn_3	Write compile output to file lfn_3.
N=lfn_4	Write new program library on file lfn_4.
S=lfn_5	Write source output on file lfn_5.
L=lfn_6	List output on file lfn_6.
LO=chars	Select list options.

char	Description
E	Errors
C	Directives other than INSERT, DELETE, RESTORE
T	Input text
M	Modifications made
W	Compile file directives
D	Deck status
S	Statistics
I	Inactive statements
A	Active statements

A	Write compressed compile file.
D	Ignore errors.
F	Modify all decks.
U	Modify only decks on DECK directives.
NR	Do not rewind compile file.
X=prog	Rewind input and output files, set A option, call program when modification is complete.
Q=prog	Rewind output file, set A option, call program assembler when modification is complete.
Z	MODIFY statement contains input directives.
CB=lfn_7	Set assembler argument B=lfn_7.
CL=lfn_8	Set assembler argument L=lfn_8.
CS=lfn_9	Set assembler argument S=lfn_9.
CG=lfn_{10}	Set assembler argument G=lfn_{10}.
CV=cs	Set character set to cs (63 or 64).

OPLEDIT($p_1, p_2, \ldots, p_n$) Removes modification decks and identifiers from a modify-formatted file.

Pi	Description
I=lfn_1	Use directive input from file lfn_1 (default is INPUT)
P=lfn_2	Use file lfn_2 for old program library (default is OPL)
N=lfn_3	Write new program library on file lfn_3 (default is NPL)
L=lfn_4	List output on file lfn_4 (default is OUTPUT)
M=lfn_5	Write output from *PULLMOD directives on file lfn_5 (if omitted, M=MODSETS assumed)
LO=x	List options:

x	Description
1	Errors
2	Directives
4	All other input statements
10_8	Modifications made
20_8	Directives processed from program library
40_8	Deck status
100_8	Directory lists
200_8	Inactive statements
400_8	Active statements

F	Modify all decks
D	Debug; ignore errors
U	Generate *EDIT directives for all decks (if omitted, generate *EDIT directives for common decks)

PROFILE(p_1, p_2, ..., p_n) Enables site to create, update, and inquire about a project profile file for user profile control.

p_i	Description
I=lfn_1	File lfn_1 contains input data (default is INPUT)
L=lfn_2	List output on file lfn_2 (default is OUTPUT)
FN=name	Indicates the family name the user wishes PROFILE to access
CN=cnum	Charge number inquire (OP=I)
PN=pnum	Project number inquire (OP=I)
CV	Convert option
OP=C	Create option
OP=K	K display option
OP=R	Restructure run
OP=S	Source run
OP=L	List option (used with LO)
OP=U	Updates project profile file
OP=T	Time-sharing update
OP=I	Inquire option
LO=F	Specifies PROFILA file
LO=C	Specifies charge numbers
LO=P	Specifies charge and project numbers
LO=FM	PROFILA file data accessible by master user
LO=CM	List of charge numbers accessible by master user
LO=PM	List of project numbers accessible by master user

/chargenum, dir_1, dir_2, . . . , dir_n Specifies PROFILE directives dir_i for charge number chargenum.

dir_i	Description
MU=mun	Master user number
M1=n	Index to SRU multiplier
M2=n	Index to SRU multiplier
M3=n	Index to SRU multiplier
M4=n	Index to SRU multiplier
AD=n	SRU constant
PN=pn	Project number
UN=un	User number
TI=ti	Time of day before which user cannot use project number
TO=to	Time of day after which user cannot use project number
CT=ct	Total connect time allowed for project number (not currently used)
AT=at	Total connect time project number has accumulated (not currently used)
SR=sr	Total SRUs allowed for project number (not currently used)
AS=as	Total SRUs project number has accumulated (not currently used)
DC=dc	Delete charge number
DP=dp	Delete project number
DU=du	Delete user number

UPDATE($p_1, p_2, \ldots, p_n$) Calls the UPDATE program.

p_i	Description
A	Sequential-to-random program library copy
B	Random-to-sequential program library copy
C=lfn_1	Write compile file output on lfn_1
D	Compile output has 80 columns for data
E	Director has actual order of decks on program library
F	Full update; all decks compiled
G=lfn_2	Output from PULLMOD written on lfn_2
I=lfn_3	Input on lfn_3
K=lfn_4	Compile output decks written on lfn_4
L=char	char specifies any of the A, F, and 0 through 9 list options
M=lfn_5	Merge input is on lfn_5
N=lfn_6	New program library written on file lfn_6
O=lfn_7	List output written on lfn_7
P=lfn_8	Use file lfn_8 for old program library
Q	Only decks on COMPILE directives processed
R=char	Files to rewind before and after update

char	Description
C	Compile
N	New program library
P	Old program library and merge library
S	Source and PULLMOD

S=lfn_9	Source output written on lfn_9
T=lfn_{10}	Source output excluding common decks on file lfn_{10}
U	Fatal errors do not halt execution
W	New program library is sequential file

p_i	Description
X	Compile file in compressed format
Z	Input file assumed in PCS compressed format
8	Compile file composed of 80-column cards
*=char	Master control character
/=char	Comment control character

PRODUCT SET CONTROL STATEMENT FORMATS

ALGOL($p_1, p_2, \ldots, p_n$) Calls ALGOL 4 compiler.

p_i	Description
A	Assembly language form of object code written on file specified by L option
B=lfn_1	Binary written on file lfn_1
C=n	Comments interpretation for special delimiters

n	Description
0	No comments interpretation
1	Debugging directives detected
2	Overlay directives detected
3	Array bound checking directives detected

D=lfn_2	Symbol file is created on file lfn_2
E	Abort job to EXIT statement for fatal errors
F	Terminate compilation after first pass if fatal error is found
I=lfn_3	Source input is on file lfn_3
K=n	Input record size. n=number of significant characters to be interpreted by compiler on source statement image
L=lfn_4	Source program listed with fatal diagnostics on file lfn_4
N	Advisory diagnostics listed on file specified by L option
O=n	Level of compiler optimization

n	Description
0	Program compiled in fast compile mode
1	Linguistic optimization
2	Subscript, statement, and O=1 opt

P=lfn_5	Assembly language punched on file lfn_5
R	Cross-reference map is produced
S=n	Array storage location: arrays in CM for n=0 and arrays in ECS for n=1
U=lfn_6	File lfn_6 contains implicit outer block

p_i	Description
X=n	Real-integer correspondence: not allowed for n=0 and allowed for n=1

BASIC($p_1, p_2, \ldots, p_n$) Calls BASIC compiler.

p_i	Description
L=lfn_1	Source, diagnostics, and execution on file lfn_1
K=lfn_2	Diagnostics and execution on file lfn_2
I=lfn_3	Source input from lfn_3
B=lfn_4	Relocatable code on file lfn_4
A=lfn_5	Assembly listing on lfn_6
N=lfn_6	Inhibit program execution

COBOL($p_1, p_2, \ldots, p_n$) Calls the COBOL compiler.

p_i	Description
A	Leading blanks treated as zeros
B=lfn_1	Object code written to file lfn_1
BUF	Minimum buffer size for version 3
C	Copy is made from source, rather than library
D	Inhibit execution when E diagnostic is encountered
DB	Check for subscript range errors
DB1	COBOL trace selected
F	Data entries described as COMPUTATIONAL-1

pi	Description
H	Increase sort efficiency
I=lfn$_2$	Compiler input obtained from file lfn$_2$
L=lfn$_3$	Output written on file lfn$_3$ The L parameter may appear with one of the following suffixes to produce special listings.

Suffix	Description
C	List of items copied from user libraries
M	Data map
O	Object code in octal
R	Data-name, procedure-name cross-reference
X	Extended diagnostics

N	Issue E diagnostic if non-ANSI feature is detected
P	Execute a strictly ANSI program
S=ulib	Satisfy external references from ulib
SUB	Suppress data division binary output
T	Request tape sort, rather than disk sort
U	Specify ASCII collating sequence
V	Save loaded program using NOGO with file name specified
Z	Ensure compatibility with version 3. Turns on C and BUF parameters

FTN($p_1, p_2, \ldots, p_n$) Calls the FORTRAN Extended compiler.

p_i	Description
A	Branch to EXIT statement if fatal compilation error occurs
B=lfn_1	Object code written on file lfn_1
BL	Generate separable output listing
C	Use COMPASS assembler for compiler code
D=lfn_2	Debug input obtained from file lfn_2
E=lfn_3	Object code on file lfn_3 output as COMPASS statement images for input to update
EL=ℓ	Diagnostic list specification

ℓ	Description
A	List fatal and non-ANSI. List informative for OPT=0, 1, or 2. List notes and warnings for TS mode.
I	List fatal. List informative for OPT=0, 1, or 2. List notes and warnings for TS mode.
W	List fatal. List warnings for TS mode.
N	List fatal. List notes and warnings for TS mode.
F	List fatal.

G=lfn_4	Load first system text overlay fron file lfn_4
GO	Binary executed after compilation
I=lfn_5	Source input is on file lfn_5
L=lfn_6	Output is written on file lfn_6
LCM=m	Address mode for level 3 data: m=D selects 17-bit address, and m=I selects 21-bit address
ML=nnn	Specifies nnn as value of MODLEVEL micro
OL	Object code listed on file specified by L
OPT=n	Level of optimization: n=0 for fast compilation, n=1 for standard compilation and execution, and n=2 for fast execution
P	Page numbering is continuous

p_i	Description
PL=n	Maximum number of records written on file specified by L
Q	Full syntactic scan performed
R=n	Reference map options n — Description 0 — No map 1 — Short map 2 — Long map 3 — Long map with common block and equivalence groups
ROUND=s	Round arithmetic operations (s=*/+ or -)
S=ovl	System text overlay loaded from library set
SEQ	Source file is in sequenced line format
SL	Source is listed on file specified by L
SYSEDIT	I/O references done indirectly through table search at object time
T	Full error traceback occurs
TS	Time-sharing mode
X=lfn_7	External text on file lfn_7
Z	Pass zero-word parameter list

SIMULA($p_1, p_2, \ldots, p_n$) Calls the SIMULA compiler.

p_i	Description
A=lfn_1	Assembly language written on file lfn_1
B=lfn_2	Assembly language written on file lfn_2
I=lfn_3	Input obtained from file lfn_3
L=lfn_4	Source input written on file lfn_4
N	Suppress array bound checking
P=lfn_5	Object code written on lfn_5 in PUNCHB format
X=lfn_6	Object code written on file lfn_6

SORTMRG ($p_1, p_2, \ldots, p_n$) Calls Sort/Merge program.

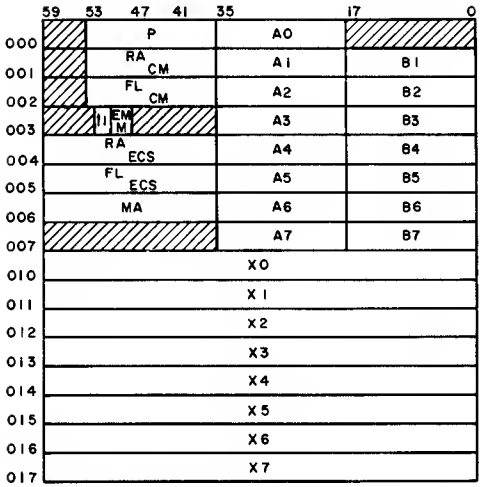
p_i	Description
I=lfn_1/r	Sort/Merge directives are on file lfn_1 with following rewind options.

r	Description
R	File is rewound before opening
NR	File is not rewound before opening

O=lfn_2/r	Listings written on file lfn_2, with rewind options listed above
OWN=lfn_3/r	Owncode binaries are located on file lfn_3, with rewind options listed above
MO=n	Intermediate merge order; $2 \leq n \leq 64$. If insufficient core is available, fatal error occurs
MO=*n	Intermediate merge order; $2 \leq n \leq 64$. If insufficient core is available, merge takes place at smaller order

SPECIAL SYSTEM INFORMATION

EXCHANGE PACKAGE AREA



P	Program address
RA	Reference address
FL	Field length
MA	Monitor address
Ai	Address registers
Bi	Increment registers
Xi	Operand registers

EM-M CPU program exit mode:

0	Disable program exit mode
1	Address out of range
2	Operand out of range
3	Address or operand out of range
4	Indefinite operand
5	Indefinite operand or address
6	Indefinite operand or operand out of range
7	Indefinite operand or address out of range or operand out of range

Ref.	Bit No.	Description
†1	52-51	Hardware error exit status bits on CYBER 70 Model 74

64-CHARACTER SET FOR TIME-SHARING TERMINALS

ASCII CODE TERMINAL†				CORRESPONDENCE CODE TERMINAL††				INTERNAL DISPLAY CODE (6/12-BIT OCTAL)
STANDARD PRINT		APL PRINT		STANDARD PRINT		APL PRINT		
CHAR.	CODE (8-BIT OCTAL)	CHAR.	CODE (8-BIT OCTAL)	CHAR.	CODE (7-BIT OCTAL)	CHAR.	CODE (7-BIT OCTAL)	
:	072	:	276	:	153	:	121	00
A	101	A	341	A	171	A	171	01
B	102	B	342	B	166	B	166	02
C	303	C	143	C	172	C	172	03
D	104	D	344	D	052	D	052	04
E	305	E	145	E	112	E	112	05
F	306	F	146	F	163	F	163	06
G	107	G	347	G	043	G	043	07
H	110	H	350	H	046	H	046	10
I	311	I	151	I	031	I	031	11
J	312	J	152	J	103	J	103	12
K	113	K	353	K	032	K	032	13
L	314	L	154	L	106	L	106	14
M	115	M	355	M	141	M	141	15
N	116	N	356	N	122	N	122	16
O	317	O	157	O	105	O	105	17
P	120	P	360	P	013	P	013	20
Q	321	Q	161	Q	133	Q	133	21
R	322	R	162	R	051	R	051	22
S	123	S	363	S	045	S	045	23
T	324	T	164	T	002	T	002	24
U	125	U	365	U	062	U	062	25
V	126	V	366	V	061	V	061	26
W	327	W	167	W	165	W	165	27
X	330	X	170	X	142	X	142	30
Y	131	Y	371	Y	147	Y	147	31
Z	132	Z	372	Z	124	Z	124	32
0	060	0	060	0	144	0	144	33
1	261	1	261	1	040	1	040	34
2	262	2	262	2	020	2	020	35
3	063	3	063	3	160	3	160	36
4	264	4	264	4	004	4	004	37
5	065	5	065	5	010	5	010	40
6	066	6	066	6	130	6	130	41
7	267	7	267	7	150	7	150	42
8	270	8	270	8	070	8	070	43
9	071	9	071	9	064	9	064	44
+	053	+	056	+	023	+	067	45
-	055	-	275	-	067	-	067	46
*	252	*	120	*	070	*	013	47
/	257	/	257	/	007	/	007	50
(	050	(	053	(	064	(	153	51
)	251	)	252	)	144	)	111	52
$	044	$	176	$	004	α	171	53
=	275	=	245	=	023	=	010	54

3AE4A

† THE OCTAL CODES LISTED FOR ASCII CODE TERMINALS ARE SHOWN WITH EVEN PARITY (NORMAL)

†† THE OCTAL CODES LISTED FOR CORRESPONDENCE CODE TERMINALS ARE SHOWN WITH OOD PARITY (NORMAL)

ASCII CODE TERMINAL				CORRESPONDENCE CODE TERMINAL				INTERNAL DISPLAY CODE (6/12-BIT OCTAL)
STANDARD PRINT		APL PRINT		STANDARD PRINT		APL PRINT		
CHAR	CODE (8-BIT OCTAL)	CHAR	CODE (8-BIT OCTAL)	CHAR	CODE (7-BIT OCTAL)	CHAR	CODE (7-BIT OCTAL)	
(SPACE)	240	(SPACE)	240	(SPACE)	100	(SPACE)	100	55
'	254	'	254	'	073	'	073	56
.	056	.	056	.	121	.	121	57
#	243	"	041	#	160	"	040	60
[	333	[	273	¼	001	[	153	61
]	335	]	072	½	001	]	111	62
%	245	÷	173	%	010	±	023	63
"	042	≠	050	"	111	#	070	64
_	137[1]	_	306	_	067	_	163	65
!	041	∨	251	≠	130	∨	064	66
&	246	∧	137	&	150	∧	144	67
'	047	'	113	'	111	'	032	70
?	077	?	321	?	007	?	133	71
<	074	<	243	NULL	——	<	160	72
>	276	>	047	NULL	——	>	150	73
@	300	≤	044	@	020	≤	004	74
\	134	\	077	NULL	——	\	007	75
^	176	¯	042	NULL	——	¯	020	76
;	273	;	074	;	153	;	073	77
\	140	NULL	——	NULL	——	NULL	——	7600
a	341	α	101	a	171	α	171	7601
b	342	⊥	102	b	166	⊥	166	7602
c	143	∩	303	c	172	∩	172	7603
d	344	L	104	d	052	L	052	7604
e	145	∈	305	e	112	∈	112	7605
f	146	X	134	f	163	X	023	7606
g	347	∇	107	g	043	∇	043	7607
h	350	Δ	110	h	046	Δ	046	7610
i	151	ι	311	i	031	ι	031	7611
j	152	∘	312	j	103	∘	103	7612
k	353	'	336	k	032	NULL	——	7613
l	154	□	314	l	106	□	106	7614
m	355	|	115	m	141	|	141	7615
n	356	⊤	116	n	122	⊤	122	7616
o	157	○	317	o	105	○	105	7617
p	360	⊢	100	p	013	⊢	001	7620
q	161	→	134	q	133	→	101	762[1]
r	162	ρ	322	r	051	ρ	051	7622
s	363	⌈	123	s	045	⌈	045	7623
t	164	~	324	t	002	~	002	7624
u	365	↓	125	u	062	↓	062	7625
v	366	∪	126	v	061	∪	061	7626
w	167	ω	327	w	165	ω	165	7627
x	170	⊃	330	x	142	⊃	142	7630
y	371	↑	131	y	147	↑	147	7631

3AE3A

1 ON TTY MODELS HAVING NO UNDERLINE, THE BACKARROW (←) TAKES ITS PLACE

ASCII CODE TERMINAL				CORRESPONDENCE CODE TERMINAL				INTERNAL DISPLAY CODE (6/12-BIT OCTAL)
STANDARD PRINT		APL PRINT		STANDARD PRINT		APL PRINT		
CHAR.	CODE (8-BIT OCTAL)	CHAR	CODE (8-BIT OCTAL)	CHAR.	CODE (8-BIT OCTAL)	CHAR.	CODE (8-BIT OCTAL)	
`	372	⊂	132	`	124	⊂	124	7632
{	173	{	140	NULL	——	NULL	——	7633
:	174	≥	246	≤	040	≥	130	7634
}	175	}	374	NULL	——	NULL	——	7635
~	176	≡	175	NULL	——	NULL	——	7636
DEL	377	DEL	377	NULL	——	NULL	——	7637
NUL	000	NUL	000	NUL	075	NUL	075	7640
SOH	201	SOH	201	SOA	166	SOA	166	7641
STX	202	STX	202	EOA	064	EOA	064	7642
ETX	003	ETX	003	NULL	——	NULL	——	7643
EOT	204	EOT	204	EOT	174	EOT	174	7644
ENQ	005	ENQ	005	NULL	——	NULL	——	7645
ACK	006	ACK	006	ACK	067	NULL	——	7646
BELL	207	BELL	207	NULL	——	NULL	——	7647
BS	210	BS	210	BS	135	BS	135	7650
HT	011	HT	011	HT	057	HT	057	7651
LF	012	LF	012	LF	156	LF	156	7652
VT	213	VT	213	NULL	——	NULL	——	7653
FF	014	FF	014	NULL	——	NULL	——	7654
CR	215	CR	215	CR	155	CR	155	7655
SO	216	SO	216	UCS	034	UCS	034	7656
SI	017	SI	017	LCS	037	LCS	037	7657
DLE	220	OLE	220	NULL	——	NULL	——	7660
DC1	021	DC1	021	NULL	——	NULL	——	7661
DC2	022	DC2	022	NULL	——	NULL	——	7662
DC3	023	DC3	023	NULL	——	NULL	——	7663
DC4	024	DC4	024	STO	054	STO	064	7664
NAK	225	NAK	225	NAK	001	NAK	001	7665
SYN	226	SYN	226	IL	075	IL	075	7666
ETB	027	ETB	027	EOB	136	EOB	136	7667
CAN	030	CAN	030	DEL	177	DEL	137	7670
EM	231	EM	231	NULL	——	NULL	——	7671
SUB	232	SUB	232	NULL	——	NULL	——	7672
ESC	033	ESC	033	PF	076	PF	076	7673
FS	234	FS	234	NULL	——	NULL	——	7674
GS	035	GS	035	NULL	——	NULL	——	7675
RS	036	RS	036	NULL	——	NULL	——	7676
US	237	US	237	NULL	——	NULL	——	7677
NULL	——	NULL	——	NULL	——	NULL	——	7400
@	300	≤	044	@	020	≤	004	7401
^	176	¯	042	NULL	——	→	020	7402
NULL	——	NULL	——	CNL	001	CNL	001	7403
NULL	——	NULL	——	NULL	——	NULL	——	7404
NULL	——	NULL	——	NULL	——	NULL	——	7405
NULL	——	NULL	——	NULL	——	NULL	——	7406
NULL	——	NULL	——	NULL	——	NULL	——	7407

3AE9A

61-CHARACTER SET FOR TIME-SHARING TERMINALS

ASCII CODE TERMINAL†				CORRESPONDENCE CODE TERMINAL††				INTERNAL DISPLAY CODE (6/12-BIT OCTAL)
STANDARD PRINT		APL PRINT		STANDARD PRINT		APL PRINT		
CHAR	CODE (8-BIT OCTAL)	CHAR	CODE (8-BIT OCTAL)	CHAR	CODE (7-BIT OCTAL)	CHAR	CODE (7-BIT OCTAL)	
NULL	——	NULL	——	NULL	——	NULL	——	00
A	101	A	341	A	171	A	171	01
B	102	B	342	B	166	B	166	02
C	303	C	143	C	172	C	172	03
D	104	D	344	D	052	D	052	04
E	305	E	145	E	112	E	112	05
F	306	F	146	F	163	F	163	06
G	107	G	347	G	043	G	043	07
H	110	H	350	H	046	H	046	10
I	311	I	151	I	031	I	031	11
J	312	J	152	J	103	J	103	12
K	113	K	353	K	032	K	032	13
L	314	L	154	L	106	L	106	14
M	115	M	355	M	141	M	141	15
N	116	N	356	N	122	N	122	16
O	317	O	157	O	105	O	105	17
P	120	P	360	P	013	P	013	20
Q	321	Q	161	Q	133	Q	133	21
R	322	R	162	R	051	R	051	22
S	123	S	363	S	045	S	045	23
T	324	T	164	T	002	T	002	24
U	125	U	365	U	062	U	062	25
V	126	V	366	V	061	V	061	26
W	327	W	167	W	165	W	165	27
X	330	X	170	X	142	X	142	30
Y	131	Y	371	Y	147	Y	147	31
Z	132	Z	372	Z	124	Z	124	32
0	060	0	060	0	144	0	144	33
1	261	1	261	1	040	1	040	34
2	262	2	262	2	020	2	020	35
3	063	3	063	3	160	3	160	36
4	264	4	264	4	004	4	004	37
5	065	5	065	5	010	5	010	40
6	066	6	066	6	130	6	130	41
7	267	7	267	7	150	7	150	42
8	270	8	270	8	070	8	070	43
9	071	9	071	9	064	9	064	44
+	053	+	055	+	023	+	067	45
-	055	-	275	-	067	-	067	46
*	252	*	120	*	070	*	013	47
/	257	/	257	/	007	/	007	50
(	050	(	053	(	064	(	153	51
)	251	)	252	)	144	)	111	52
$	044	$	176	$	004	α	171	53
=	275	=	245	=	023	=	010	54

5AE4A

† THE OCTAL CODES LISTED FOR ASCII CODE TERMINALS ARE SHOWN WITH EVEN PARITY (NORMAL)

†† THE OCTAL CODES LISTED FOR CORRESPONDENCE CODE TERMINALS ARE SHOWN WITH ODD PARITY (NORMAL)

ASCII CODE TERMINAL				CORRESPONDENCE CODE TERMINAL				INTERNAL DISPLAY CODE (6/12-BIT OCTAL)
STANDARD PRINT		APL PRINT		STANDARD PRINT		APL PRINT		
CHAR	CODE (8-BIT OCTAL)	CHAR	CODE (8-BIT OCTAL)	CHAR	CODE (7-BIT OCTAL)	CHAR	CODE (7-BIT OCTAL)	
(SPACE)	240	(SPACE)	240	(SPACE)	100	(SPACE)	100	55
,	254	,	254	,	073	,	073	56
.	056	.	056	.	121	.	121	57
"	042	"	041	"	111	⍞	040	60
[	333	[	273	⅛	001	[	133	61
]	335	]	072	⅛	001	]	111	62
:	072	:	276	:	163	:	121	63
'	047	'	113	'	111	'	032	64
⊗	246	×	164	⊗	160	×	023	65
CR	215	CR	215	NULL	——	NULL	——	66
LF	012	LF	012	LF	156	LF	156	67
↑	336	¯	042	NULL	——	¯	020	70
#	243	≠	336	#	160	≠	070	71
<	074	<	243	NULL	——	<	160	72
>	276	>	047	NULL	——	>	150	73
(ESC 1)	——	NULL	——	NULL	——	NULL	——	74
?	077	?	321	?	007	?	133	75
(ESC 2)	——	NULL	——	NULL	——	NULL	——	76
;	273	;	074	;	153	;	073	77
NULL	——	NULL	——	NULL	——	NULL	——	7600
a	341	α	101	a	171	α	171	7601
b	342	⊥	102	b	166	⊥	166	7602
c	143	∩	303	c	172	∩	172	7603
d	344	L	104	d	052	L	052	7604
e	145	∈	305	e	112	∈	112	7605
f	146	∧	137	f	163	_	163	7606
g	347	∇	107	g	043	∇	043	7607
h	350	Δ	110	h	046	Δ	046	7610
i	151	ι	311	i	031	ι	031	7611
j	152	○	312	j	103	○	103	7612
k	353	←	100	k	032	≤	004	7613
l	154	□	314	l	106	□	106	7614
m	355	→	134	m	141	I	141	7615
n	356	⊤	116	n	122	⊤	122	7616
o	157	○	317	o	105	○	105	7617
p	360	≤	044	p	013	≥	130	7620
q	161	≥	246	q	133	?	133	7621
r	162	⍴	322	r	051	⍴	051	7622
s	363	⌈	123	s	045	⌈	045	7623
t	164	≠	050	t	002	~	002	7524
u	365	↓	125	u	D62	↓	062	7625
v	366	∪	126	v	061	∪	061	7626
w	167	ω	327	w	165	ω	165	7627
x	170	⊃	330	x	142	⊃	142	7630
y	371	↑	131	y	147	↑	147	7631

3AE3A

ASCII CODE TERMINAL				CORRESPONDENCE CODE TERMINAL				INTERNAL DISPLAY CODE (6/12-BIT OCTAL)
STANDARD PRINT		APL PRINT		STANDARD PRINT		APL PRINT		
CHAR	CODE (8-BIT OCTAL)	CHAR	CODE (8-BIT OCTAL)	CHAR	CODE (8-BIT OCTAL)	CHAR	CODE (8-BIT OCTAL)	
z	372	⊂	132	z	124	⊂	124	7632
DLE	220	OLE	220	NULL	——	NULL	——	7633
BELL	207	BELL	207	NULL	——	NULL	——	7634
DC2	022	0C2	022	NULL	——	NULL	——	7635
ETX	003	ETX	003	NULL	——	NULL	——	7636
0C4	024	DC4	024	NULL	——	NULL	——	7637
NAK	225	NAK	225	NULL	——	NULL	——	7640
SYN	226	SYN	226	NULL	——	NULL	——	7641
ETB	027	ETB	027	NULL	——	NULL	——	7642
CAN	030	CAN	030	NULL	——	NULL	——	7643
EM	231	EM	231	NULL	——	NULL	——	7644
VT	213	VT	213	NULL	——	NULL	——	7645
SOH	201	SOH	201	NULL	——	NULL	——	7646
!	041	V	251	NULL	——	NULL	——	7647
SI	017	SI	017	NULL	——	NULL	——	7650
BS	210	BS	210	BS	135	BS	135	7651
HT	011	HT	011	HT	057	NT	057	7652
EOT	204	EOT	204	NULL	——	NULL	——	7653
GS	035	GS	035	NULL	——	NULL	——	7654
NUL	000	NUL	000	NUL	075	NUL	075	7655
FF	014	FF	014	,	073	NULL	——	7656
SO	216	SO	216	.	121	NULL	——	7657
STX	202	STX	202	NULL	——	NULL	——	7660
{	173	{	140	NULL	——	←	001	7661
}	175	}	374	NULL	——	→	101	7662
SUB	232	SUB	232	NULL	——	NULL	——	7663
ACK	006	ACK	006	NULL	——	NULL	——	7664
&	246	NULL	——	NULL	——	NULL	——	7665
\	134	\	077	NULL	——	\	007	7666
:	174	I	115	↑	130	I	141	7667
~	176	~	324	±	040	~	002	7670
#	243	NULL	——	NULL	——	NULL	——	7671
FS	234	FS	234	NULL	——	NULL	——	7672
RS	036	RS	036	NULL	——	NULL	——	7673
DEL	377	DEL	377	NULL	——	NULL	——	7674
US	237	US	237	NULL	——	NULL	——	7675
NL	——	NL	——	NL	155	NL	155	7676
ESC	033	ESC	033	NULL	——	NULL	——	7677
NULL	——	NULL	——	NULL	——	NULL	——	7400
@	300	≡	175	@	020	Λ	144	7401
%	245	÷	173	%	010	÷	023	7402
\	140	≠	335	NULL	——	V	064	7403
—	137 †	—	306	—	067	NULL	——	7404
X-ON	021	X-ON	021	NULL	——	NULL	——	7405
X-OFF	223	X-OFF	223	NULL	——	NULL	——	7406
ENQ	005	ENQ	005	NULL	——	NULL	——	7407

3AE8A

† ON TTY MOOELS HAVING NO UNOERLINE, THE BACKARROW (—) TAKES ITS PLACE.

STANDARD CHARACTER SET

CDC GRAPHIC	ASCII GRAPHIC SUBSET	DISPLAY CODE	HOLLERITH PUNCH (026)	EXTERNAL BCD CODE	ASCII PUNCH (029)	ASCII CODE
:†	:	00†	B-2	00	8-2	3A
A	A	01	12-1	61	12-1	41
B	B	02	12-2	62	12-2	42
C	C	03	12-3	63	12-3	43
D	D	04	12-4	64	12-4	44
E	E	05	12-5	65	12-5	45
F	F	06	12-6	66	12-6	46
G	G	07	12-7	67	12-7	47
H	H	10	12-8	70	12-8	48
I	I	11	12-9	71	12-9	49
J	J	12	11-1	41	11-1	4A
K	K	13	11-2	42	11-2	4B
L	L	14	11-3	43	11-3	4C
M	M	15	11-4	44	11-4	4D
N	N	16	11-5	45	11-5	4E
O	O	17	11-6	48	11-8	4F
P	P	20	11-7	47	11-7	50
Q	Q	21	11-8	50	11-8	51
R	R	22	11-9	51	11-9	52
S	S	23	0-2	22	0-2	53
T	T	24	0-3	23	0-3	54
U	U	25	0-4	24	0-4	55
V	V	26	0-5	25	0-5	56
W	W	27	0-8	26	0-6	57
X	X	30	0-7	27	0-7	58
Y	Y	31	0-8	30	0-8	59
Z	Z	32	0-9	31	0-9	5A
0	0	33	0	12	0	30
1	1	34	1	01	1	31
2	2	35	2	02	2	32
3	3	36	3	03	3	33
4	4	37	4	04	4	34
5	5	40	5	05	5	35

3AE13A

† TWELVE OR MORE ZERO BITS AT THE END OF A 60-BIT WORD ARE INTERPRETED AS END-OF-LINE MARK RATHER THAN TWO COLONS. END-OF-LINE MARK IS CONVERTED TO EXTERNAL BCD 1632.

CDC GRAPHIC	ASCII GRAPHIC SUBSET	DISPLAY CODE	HOLLERITH PUNCH (026)	EXTERNAL BCD CODE	ASCII PUNCH (029)	ASCII CODE
6	6	41	6	06	6	36
7	7	42	7	07	7	37
8	8	43	8	10	8	38
9	9	44	9	11	9	39
+	+	45	12	60	12-8-6	28
-	-	46	11	40	11	2D
*	*	47	11-8-4	54	11-8-4	2A
/	/	50	0-1	21	0-1	2F
(	(	51	0-8-4	34	12-8-5	28
)	)	52	12-8-4	74	11-8-5	29
$	$	53	11-8-3	53	11-8-3	24
=	=	54	8-3	13	8-8	3D
BLANK	BLANK	55	NO PUNCH	20	NO PUNCH	20
,(COMMA)	,(COMMA)	58	0-8-3	33	0-8-3	2C
.(PERIOD)	.(PERIOD)	57	12-8-3	73	12-8-3	2E
≡	#	60	0-8-6	38	8-3	23
[	[	81	8-7	17	12-8-2	58
]	]	62	0-8-2	32	11-8-2	5D
%††	%	63	8-8	18	0-8-4	25
≠	"(QUOTE)	64	8-4	14	8-7	22
←	_(UNDERLINE)	65	0-8-5	35	0-8-5	5F
∨	!	68	11-0	52	12-8-7	21
∧	&	67	0-8-7	37	12	26
↑	'(APOSTROPHE)	70	11-8-5	55	8-5	27
↓	?	71	11-8-6	56	0-8-7	3F
<	<	72	12-0	72	12-8-4	3C
>	>	73	11-6-7	57	0-8-6	3E
≤	@	74	8-5	15	8-4	40
≥	\	75	12-8-5	75	0-8-2	5C
¬	~(CIRCUMFLEX)	78	12-8-8	76	11-8-7	5E
; (SEMICOLON)	; (SEMICOLON)	77	12-8-7	77	11-8-6	3B

3AE6A

†† IN INSTALLATIONS USING THE CDC 63-GRAPHIC SET, DISPLAY CODE 00 HAS NO ASSOCIATED GRAPHIC OR HOLLERITH CODE; DISPLAY CODE 63 IS THE COLON(8-2 PUNCH). THE SELECTION OF THE 63- OR 64-CHARACTER SET FOR TAPES IS AN INSTALLATION OPTION.

ASCII/DISPLAY CODE AND EBCDIC/DISPLAY CODE CONVERSION

DISPLAY CODE		ASCII				EBCDIC			
		UPPERCASE		LOWERCASE		UPPERCASE		LOWERCASE	
OCTAL	CHAR	CHAR	HEX	CHAR	HEX	CHAR	HEX	CHAR	HEX
00	:	:	3A	SUB	1A	:	7A	SUB	3F
01	A	A	41	a	61	A	C1	a	81
02	B	B	42	b	62	B	C2	b	82
03	C	C	43	c	63	C	C3	c	83
04	D	D	44	d	64	D	C4	d	84
05	E	E	45	e	65	E	C5	e	85
06	F	F	46	f	66	F	C6	f	86
07	G	G	47	g	67	G	C7	g	87
10	H	H	48	h	68	H	C8	h	88
11	I	I	49	i	69	I	C9	i	89
12	J	J	4A	j	6A	J	D1	j	91
13	K	K	4B	k	6B	K	D2	k	92
14	L	L	4C	l	6C	L	D3	l	93
15	M	M	4D	m	6D	M	D4	m	94
16	N	N	4E	n	6E	N	D5	n	95
17	O	O	4F	o	6F	O	D6	o	96
20	P	P	50	p	70	P	D7	p	97
21	Q	Q	51	q	71	Q	D8	q	98
22	R	R	52	r	72	R	D9	r	99
23	S	S	53	s	73	S	E2	s	A2
24	T	T	54	t	74	T	E3	t	A3
25	U	U	55	u	75	U	E4	u	A4
26	V	V	56	v	76	V	E5	v	A5
27	W	W	57	w	77	W	E6	w	A6
30	X	X	58	x	78	X	E7	x	A7
31	Y	Y	59	y	79	Y	E8	y	A8
32	Z	Z	5A	z	7A	Z	E9	z	A9
33	0	0	30	OLE	10	0	F0	OLE	10
34	1	1	31	DC1	11	1	F1	DC1	11
35	2	2	32	DC2	12	2	F2	DC2	12
36	3	3	33	DC3	13	3	F3	TM	13
37	4	4	34	DC4	14	4	F4	DC4	3C

3AE7A

DISPLAY CODE		ASCII				EBCDIC			
		UPPERCASE		LOWERCASE		UPPERCASE		LOWERCASE	
OCTAL	CHAR	CHAR	HEX	CHAR	HEX	CHAR	HEX	CHAR	HEX
40	5	5	35	NAK	15	5	F5	NAK	3D
41	6	6	36	SYN	16	6	F6	SYN	32
42	7	7	37	ETB	17	7	F7	ETB	26
43	8	8	38	CAN	18	8	F8	CAN	18
44	9	9	39	EM	19	9	F9	EM	19
45	+	+	2B	VT	0B	+	4E	VT	0B
46	-	-	2D	CR	0D	-	60	CR	0D
47	*	*	2A	LF	0A	*	5C	LF	25
50	/	/	2F	SI	0F	/	61	SI	0F
51	(	(	28	BS	08	(	4D	BS	16
52	)	)	29	HT	09	)	5D	HT	05
53	$	$	24	EOT	04	$	5B	EOT	37
54	=	=	30	GS	1D	=	7E	IGS	1D
55	SP	SP	20	NUL	00	SP	40	NUL	00
56	,	,	2C	FF	0C	,	6B	FF	0C
57	.	.	2E	SO	0E	.	4B	SO	0E
60	≡	#	23	ETX	03	#	7B	ETX	03
61	[	[	5B	FS	1C	¢	4A	IFS	1C
62	]	]	50	SOH	01	!	5A	SOH	01
63	%	%	25	ENQ	05	%	6C	ENQ	2D
64	≠	"	22	STX	02	"	7F	STX	02
65	↑	_	5F	DEL	7F	_	6D	DEL	07
66	∨	!	21	}	7D	|	4F	}	D0
67	∧	&	26	ACK	06	&	50	ACK	2E
70	↑	'	27	BEL	07	'	7D	BEL	2F
71	↓	?	3F	US	1F	?	6F	IUS	1F
72	<	<	3C	{	7B	<	4C	{	C0
73	>	>	3E	RS	1E	>	6E	IRS	1E
74	≤	@	40	`	60	@	7C	`	79
75	≥	\	5C	;	7C	\	E0	:	6A
76	¬	^	5E	~	7E	¬	5F	~	A1
77	;	;	3B	ESC	1B	;	5E	ESC	27

3AE8A

NOTES

1. Uppercase and lowercase apply only to the case conversions and do not necessarily reflect any true case.
2. When translating from display code to ASCII/ EBCDIC, the uppercase equivalent character is taken.
3. When translating from ASCII/EBCDIC to display code, the uppercase and lowercase characters fold together to a single display code equivalent character.
4. All EBCDIC codes not used are translated to display code 55 (SP).
5. If a 9-track tape is read with ASCII conversion and a character value above $7F_{16}$ is encountered, a flag word error is given.
6. In a 63-character set system, the display code for the : graphic is 63. The % character does not exist and ASCII/EBCDIC % or ENQ are translated to display code 55.